IN THE CLAIMS

Amended claims follow:

1. – 44. (Cancelled)

45. (Currently Amended) A method for programmable processing in a hardware graphics accelerator, comprising:

receiving graphics data including lighting information in a hardware graphics accelerator; and

performing programmable operations on the graphics data utilizing the hardware graphics accelerator in order to generate output, wherein the operations are programmable by a user utilizing instructions from a predetermined instruction set capable of being executed by the hardware graphics accelerator;

wherein the operations include a set on less operation, a move operation, a multiply operation, an addition operation, a multiply and addition operation, a reciprocal operation, a reciprocal square root operation, a three component dot product operation, a four component dot product operation, a distance operation, a minimum operation, a maximum operation, a set on greater or equal than operation, an exponential operation, a logarithm operation, and a lighting operation.

46. – 47. (Cancelled)

48. (Currently Amended) A method for processing graphics data, comprising:

transforming the graphics data utilizing a hardware graphics accelerator; and

lighting the graphics data utilizing the hardware graphics accelerator;

wherein at least one of the transforming and the lighting includes performing operations on the graphics data utilizing instructions from an instruction set capable of being executed by the hardware graphics accelerator, the operations including a ~~no operation, a load, a move, a multiply, an addition, and a set on less than each capable of being carried out by the hardware graphics accelerator~~ reciprocal operation, a reciprocal

square root operation, a three component dot product operation, a four component dot product operation, a distance operation, a minimum operation, a maximum operation, an exponential operation, and a logarithm operation.

49. (Currently Amended) A method for processing graphics data, comprising:

transforming the graphics data utilizing a hardware graphics accelerator; and

lighting the graphics data utilizing the hardware graphics accelerator;

wherein at least one of the transforming and the lighting includes performing operations on the graphics data utilizing instructions from an instruction set capable of being executed by the hardware graphics accelerator, the instruction set including a ~~no operation instruction, a load instruction, a move instruction, a multiply instruction, an addition instruction, and a set on less than instruction~~multiply and addition instruction, a reciprocal instruction, a reciprocal square root instruction, a three component dot product instruction, a four component dot product instruction, a distance instruction, a minimum instruction, a maximum instruction, a set on greater or equal than instruction, an exponential instruction, a logarithm instruction, and a lighting instruction.

50. – 51. (Cancelled)

52. (Currently Amended) A method for processing graphics data utilizing a hardware graphics accelerator, comprising:

transforming the graphics data utilizing the hardware graphics accelerator; and

lighting the graphics data utilizing the hardware graphics accelerator;

wherein the transforming and the lighting include performing operations on the graphics data utilizing instructions from an instruction set capable of being executed by the hardware graphics accelerator, the instruction set including a no operation instruction, a load instruction, a move instruction, a multiply instruction, an addition instruction, [[and ]]a set on less than instruction, a multiply and addition instruction, a reciprocal instruction, a reciprocal square root instruction, a three component dot product instruction, a four component dot product instruction, a distance instruction, a minimum

instruction, a maximum instruction, a set on greater or equal than instruction, an exponential instruction, a logarithm instruction, and a lighting instruction;

wherein the transforming and the lighting further include negating the graphics data and branching.

53. (Currently Amended) A method for processing graphics data utilizing a hardware graphics accelerator, comprising:

transforming the graphics data utilizing the hardware graphics accelerator, the graphics data including constants; and

lighting the graphics data utilizing the hardware graphics accelerator;

wherein the transforming and the lighting include performing operations on the graphics data utilizing instructions from an instruction set capable of being executed by the hardware graphics accelerator, the instruction set including a ~~no-operation instruction, a load instruction, a move instruction, a multiply instruction, an addition instruction, and a set on less than instruction~~multiply and addition instruction, a reciprocal instruction, a reciprocal square root instruction, a three component dot product instruction, a four component dot product instruction, a distance instruction, a minimum instruction, a maximum instruction, a set on greater or equal than instruction, an exponential instruction, a logarithm instruction, and a lighting instruction;

wherein the transforming and the lighting further include ~~negating the graphics data and~~ branching;

wherein a plurality of the operations are performed in parallel;

wherein the hardware graphics accelerator operates with an OpenGL application program interface.

54. (Cancelled)

55. (Previously Presented) A method as recited in claim 45, wherein the graphics data includes vertex data, and the operations perform vertex processing on the vertex data.

56. (Previously Presented) A method as recited in claim 55, wherein multiple vertices represented by the vertex data are operated upon in parallel.

57. (Previously Presented) A method as recited in claim 45, wherein the graphics data is swizzled.

58. (Cancelled)

59. (Previously Presented) A method as recited in claim 52, wherein the graphics data includes vertex data, and the operations perform vertex processing on the vertex data.

60. (Previously Presented) A method as recited in claim 59, wherein multiple vertices represented by the vertex data are operated upon in parallel.

61. (Previously Presented) A method as recited in claim 52, wherein the graphics data is swizzled.

62. (Previously Presented) A method as recited in claim 52, and further comprising:

determining whether the hardware graphics accelerator is operating in a programmable mode;

performing the operations on the graphics data if it is determined that the hardware graphics accelerator is operating in the programmable mode; and

operating on the graphics data in accordance with a standard graphics application program interface if it is determined that the hardware graphics accelerator is not operating in the programmable mode.

63. (New) A hardware graphics accelerator for receiving graphics data, and performing programmable operations on the graphics data in order to generate output, wherein the operations are programmable by a user utilizing instructions from a predetermined instruction set capable of being executed by the hardware graphics accelerator, the predetermined instruction set including a reciprocal instruction, a

reciprocal square root instruction, a three component dot product instruction, a four component dot product instruction, a distance instruction, a minimum instruction, a maximum instruction, an exponential instruction, and a logarithm instruction.

64. (New) A system, comprising:

a central processing unit; and

a hardware graphics accelerator for receiving graphics data, and performing programmable operations on the graphics data in order to generate output;

wherein the operations are programmable by a user utilizing instructions from a predetermined instruction set capable of being executed by the hardware graphics accelerator, the predetermined instruction set including a reciprocal instruction, a reciprocal square root instruction, a three component dot product instruction, a four component dot product instruction, a distance instruction, a minimum instruction, a maximum instruction, an exponential instruction, and a logarithm instruction.